

一、 名词解释 (每小题 2 分)

1. 地址映射: 将逻辑地址转换为物理地址
2. 动态重定位: 根据实际分配的内存位置对地址进行调整
3. 虚拟存储器: 将内存和外存结合起来提供一个更大的地址空间
4. 静态链接: 在程序编译和加载过程中, 将各个目标模块组合成一个可执行文件
5. 对换: 将暂时用不到的进程从内存移到外存的过程
6. 设备驱动程序: 操作系统与硬件设备之间的软件接口
7. SPooling: 基于 I/O 缓冲区的输入输出技术, 用于设备间的并行操作
8. I/O 通道: 专门处理输入输出的硬件设备
9. 文件系统: 用于管理和组织文件和目录的软件和数据结构的集合
10. 目标文件: 编译器生成的二进制文件, 包含程序的机器代码和符号信息
11. 文件的逻辑结构: 用户从逻辑上看到的文件组织形式
12. 有结构文件: 文件中的数据具有明确的组织形式和结构
13. 位示图: 管理磁盘空间与内存空间的数据结构
14. 程序接口: 程序与外部环境交互的接口
15. 系统调用: 用户程序请求操作系统的一种方式
16. I/O 中断: I/O 设备向 CPU 发出的中断/请求信号
17. 文件管理系统: 操作系统中负责管理文件和目录的子系统
18. 文件: 存储在存储介质上的一组相关信息的集合
19. 文件的逻辑结构: 同
20. 文件的物理结构: 文件存储在物理存储介质上的实际结构

二、 填空题 (每空 1 分)

1. 程序被装入内存时由操作系统的连接装入程序完成程序的逻辑地址到内存地址的转换, 也称为 静态重定位。
2. 程序的装入方式包括 绝对装入、动态相对装入 和 静态相对装入 三种。
3. 程序的链接方式包括 静态链接、装入时动态链接 和 运行时尚态链接 三种。
4. 在分区管理方式中, 空闲分区的管理所使用的数据结构包括 空闲分区表 和 空闲分区链。
5. 将系统中所有空闲的小分区集中起来形成一个大分区的过程称为 拼接。
6. 比较分页与分段管理, 页的大小是 固定 的, 由 系统 决定, 而段的大小是 灵活 的, 由 用户程序 决定。
7. 虚拟存储器的主要特征包括多次性、对换性 和 虚拟性。
8. 文件系统由 文件存储空间、目录管理 和 文件操作 部分组成。
9. 内存的动态分区分配方式涉及: 分区分配、分区保护、分区回收 这三方面的问题。

10. 内存连续分配方式容易产生碎片，从而降低内存的使用率。
11. 内存的离散分配方式大致可分为：固定分区分配方式、动态分区分配方式、非连续分配方式。
12. 在内存的离散分配管理方式下，为了提高地址变换速度，可增设一种特殊的缓冲寄存器，称为快表。
13. 分页存储管理将进程的虚拟地址空间和物理地址空间分为相同大小的页面。
14. 分段存储管理方式的优点包括：逻辑结构清晰、便于程序员编写、动态链接和动态支持地址变换、减少内存碎片。
15. 地址映射实现进程从逻辑地址到物理地址的变换功能。
16. 程序运行时对内存访问存在局部性现象，表现为时间局部性和空间局部性。
17. I/O 通道是一种特殊的处理机，但与其差别在于 (1) 专门用于 I/O 操作 (2) I/O 通道操作更简单。
18. 有结构文件的组织方式可大致分为 记录式文件、索引文件 和 流式文件 三大类。
19. 典型的磁盘索引组织方式包括 单链索引、多链索引 和 索引索引。
20. 在作业执行期间才进行的地址变换方式是 动态重定位。
21. I/O 管理系统软件可以分为三个层次：用户程序、设备无关、设备驱动程序。
22. 在顺序结构、链接结构和索引结构三种文件的物理结构中，随机访问效率最高的是索引结构。
23. 计算机系统存储层次大致可分为寄存器、主存储器和辅助存储器层次。
24. 计算机主存分配存储管理方式可分为连续分配和非连续分配方式。
25. 文件的物理结构不仅与存储特性有关，而且与文件系统设计有关。
26. 根据记录的组织方式，可把文件的逻辑结构分为顺序文件、索引文件和流式文件三大类。
27. 外存的分配方式有连续分配、链接分配和索引分配三大类。
28. 在树型目录结构中，根据路径的起点不同，可把路径分为绝对路径和相对路径两种。
29. 在采用空闲链表法来管理空闲盘区时，有顺序链接和索引链接两种形式。

30. 文件的共享分为共享文件系统和分布式文件系统两种方式。
31. 按照信息交换的单位可把设备分为块设备和字符设备两大类;而按照设备的共享属性又可把设备分为独占设备、共享设备和虚拟设备三大类。
32. I/O 设备的控制方式可分为程序控制方式、中断驱动方式、中断方式和 DMA 等。
33. 为了缓冲 CPU 与 I/O 设备速度不匹配的矛盾,在 CPU 和 I/O 设备之间引入了缓冲技术,缓冲可分为单缓冲、双缓冲、循环缓冲和缓冲池四种。
34. 磁盘访问时间包括寻道时间、旋转延迟时间和数据传输时间

三、 简答题 (每小题 6 分)

1. 计算机存储器系统主要有哪些层次?各个层次又包含哪些内容?试从存储性能角度对存储系统进行分析。
2. 简述计算机程序从代码编写到运行完毕所经历的主要过程。
3. 程序链接的主要方式有哪些?它们的异同点是什么?
4. 内存动态分区分配算法根据搜索空闲区的方式可分为哪些类型,它们又分别有哪些典型的分配算法?请逐一简述这些算法的核心思想。
5. 列举典型的内存动态分区算法和 CPU 调度算法,并分析两者之间的异同点。
6. 列举典型的页面置换算法和 CPU 调度算法,并分析两者之间的异同点。
7. 列举典型的页面置换算法和磁盘调度算法,并分析两者之间的异同点。
8. 分别简述分页存储、分段存储和段页存储的地址变换过程。
9. 简述中断处理程序的各个处理步骤。
10. 从用途、数据类型、组织和管理方式等角度简述文件类型分类。
11. 请详述文件目录的分类及相应查询方式。
12. 请简述文件结构的三种主要组织方式,并对比分析各自优劣。
13. 请分别简述内存和外存的存储分配空间分配方式,并对比分析它们之间的异同点。
14. 请分别简述提高磁盘 I/O 速度的多种途径。

1. 计算机存储器系统主要层次、内容及性能分析

- **层次及内容**：计算机存储器系统主要有高速缓冲存储器（Cache）、主存储器（内存）、辅助存储器（如硬盘、光盘等）及离线存储（如磁带库等）这几个层次。其中，Cache 是位于 CPU 和主存之间的高速小容量存储器，用于暂存主存中近期常用的数据和指令，以加快 CPU 的访问速度；主存储器是计算机的主要存储设备，用于存储当前正在运行的程序和数据；辅助存储器用于长期存储大量数据，其存储容量大但存取速度较慢；离线存储一般用于长期备份数据，通常需要人工干预进行数据读写。
- **性能分析**：从存储性能角度看，Cache 的存取速度最快，但容量小，其命中率对系统性能影响大，若命中率高，则可有效提升整个存储系统的速度；主存的存取速度次之，但容量相对较大，能满足一般程序运行时的数据存储需求；辅助存储器的存取速度较慢，不过容量巨大，能在成本相对较低的情况下提供大量数据的存储空间，但其数据读写会受到机械运动等因素影响，速度成为系统性能瓶颈之一；离线存储的存取速度最慢，但在数据安全性及长期大量数据备份方面有重要意义。整个存储系统的性能是在速度、容量和成本等多方面因素综合权衡下实现的，通过层次结构的设计，在保证一定速度的前提下，利用大容量的辅助存储和离线存储来满足数据存储的需求。

2. 计算机程序从代码编写到运行完毕的主要过程

- 程序员使用高级语言（如 C、Java 等）编写源代码，形成源程序文件。
- 通过编译器对源程序进行编译，将源代码转换为目标代码，生成目标文件，此过程会进行语法检查等，若有错误会提示程序员修改。
- 经过链接器将多个目标文件以及所需的库函数等链接在一起，形成一个可执行文件，该文件包含了完整的程序代码和数据信息，能在操作系统上运行。
- 操作系统加载器将可执行文件从磁盘加载到内存中，为其分配所需的内存空间，并进行一系列初始化操作，使程序处于可运行状态。
- CPU 执行程序中的指令，程序开始运行，在运行过程中会与输入输出设备等进行交互，处理数据，完成相应的任务。
- 当程序运行结束，或遇到结束指令、异常等情况时，会进行清理工作，释放所占用的资源，如内存空间等，程序运行完毕。

3. 程序链接的主要方式及异同点

- **主要方式**：有静态链接和动态链接。静态链接是在程序运行前，将所有目标模块及所需库函数等链接成一个完整的可执行文件，之后不再发生变化；动态链接则是在程序装入内存或运行时才链接所需的模块，通常以动态链接库（DLL）的形式存在，多个程序可共享一个动态链接库。
- **相同点**：都是将程序运行所需的各个分离的代码模块组合成一个可执行的程序，以确保程序能够正确运行，且都需要链接器等相关工具来完成链接操作。
- **不同点**：静态链接在编译时就将所有模块链接在一起，生成一个独立的可执行文件，程序运行时不依赖外部模块，但生成的可执行文件较大，且更新模块不方便；动态链接在程序运行时装入或调用时才链接相应模块，生成的可执行文件较小，能实现多个程序共享代码，利于软件的更新和维护，但程序运行时需要链接库的支持，若链接库缺失或版本不匹配可能会出现问題。

4. 内存动态分区分配算法的类型及典型算法

• 类型及典型算法

- **首次适应算法（First Fit）**：从内存分区的开始部分依次查找空闲区，找到第一个能满足作业要求的空闲区后，便将该空闲区分配给作业。其思想是先找到一个合适的空闲区就停止查找并分配，优点是算法简单，空闲区分配较为均匀，避免了某些空闲区一直被占用而后面大量小空闲区无法利用的情况，但可能会导致内存碎片较多且前面的空闲区较大，后面的空闲区较小且零散。
- **循环首次适应算法（Next Fit）**：类似于首次适应算法，但不是从内存分区的开头开始查找，而是从上一次找到的空闲区的下一个位置开始查找，直至绕回到内存分区的最前面。这种方法减小了首次适应算法中对低地址部分空闲区的频繁扫描，提高了查找效率，但同样存在内存碎片较多的问题，且分配可能不如首次适应算法均匀。
- **最佳适应算法（Best Fit）**：在整个空闲区列表中寻找能够满足作业要求且大小最小的空闲区分配给作业。其核心是尽可能充分利用小空闲区，减少大空闲区的浪费，但容易造成一些较大的空闲区无法被有效利用，且随着内存分配和释放的进行，可能产生越来越多的小碎片。
- **最坏适应算法（Worst Fit）**：在整个空闲区列表中寻找能够满足作业要求且大小最大的空闲区分配给作业。其想法是把大块留到后面用，避免频繁分配大块导致后续无法满足大的作业需求，但这样可能会产生较多的大碎片，降低内存的利用效率。

5. 典型的内存动态分区算法和 CPU 调度算法及异同点

内存动态分区算法：

- 首次适应算法（First Fit）
- 循环首次适应算法（Next Fit）
- 最佳适应算法（Best Fit）
- 最差适应算法（Worst Fit）

CPU 调度算法：

- 先来先服务（FCFS）
- 短作业优先（SJF）
- 优先级调度
- 时间片轮转（RR）
- 多级队列调度

异同点：

- 相同点：两者均为资源分配策略，目标是高效利用系统资源（内存或 CPU）。
- 异同点：
 - 动态分区算法解决内存的连续分配问题，需处理外部碎片；而 CPU 调度算法关注进程执行顺序，不涉及物理空间连续性问题。
 - 动态分区算法需合并空闲区以减少碎片，而 CPU 调度算法侧重公平性和响应时间优化。

6. 典型的页面置换算法和 CPU 调度算法及异同点

页面置换算法：

- 最佳置换算法（OPT）
- 先进先出（FIFO）
- 最近最久未使用（LRU）
- Clock 算法
- 最少使用（LFU）

CPU 调度算法：同上（见问题 5）

异同点：

- 相同点：均基于队列管理思想，需权衡效率与实现复杂度。例如，FIFO 和 LRU 在两者中均有类似逻辑（如 FIFO 页面置换与 FCFS 调度）。

- 不同点：
 - 页面置换算法用于解决内存不足时的页面替换问题，关注局部性原理；而 CPU 调度算法决定进程执行顺序，关注公平性和吞吐量。
 - 页面置换算法需预测未来访问模式（如 OPT），而 CPU 调度算法更依赖当前运行状态（如优先级）。

7. 典型的页面置换算法和磁盘调度算法及异同点

页面置换算法：同上（见问题 6）

磁盘调度算法：

- 先来先服务（FCFS）
- 最短寻道时间优先（SSTF）
- 扫描算法（SCAN）
- 循环扫描（C-SCAN）
- N-step-SCAN [[资料未直接提及，通用知识补充]]

异同点：

- 相同点：两者均需优化访问顺序以减少开销（页面置换减少缺页率，磁盘调度减少寻道时间）。
- 不同点：
 - 页面置换算法针对内存与磁盘间的数据交换，依赖程序局部性；而磁盘调度算法直接优化磁盘物理访问路径。
 - 页面置换算法（如 LRU）需跟踪访问历史，而磁盘调度算法（如 SCAN）需动态调整磁头移动方向。

8. 分页、分段和段页存储的地址变换过程

分页存储：

- 逻辑地址分为页号和页内偏移量，通过页表将页号映射到物理块号，最终地址 = 物理块号 × 块大小 + 页内偏移量 [[通用知识]]。

分段存储：

- 逻辑地址由段号和段内偏移量组成，通过段表获取段基址，最终地址 = 段基址 + 段内偏移量 [[通用知识]]。

段页存储：

- 先将逻辑地址划分为段号和段内页号，再通过段表找到页表基址，最终地址由页表项中的物理块号与页内偏移量计算得出 [[通用知识]]。

9. 中断处理程序的处理步骤

1. 保存上下文：保存被中断进程的寄存器状态。
2. 识别中断源：确定中断类型（如 I/O 完成、硬件信号）。
3. 执行中断服务例程：处理中断事件（如数据读取）。
4. 恢复上下文：恢复被中断进程的状态或调度新进程。
5. 返回中断点：继续执行被中断的指令流 [[通用知识]]。

10. 文件类型的分类（用途、数据类型、组织和管理方式）

- 用途：系统文件（如配置文件）、用户文件（如文档）、库文件（如代码库）。
- 数据类型：文本文件（ASCII）、二进制文件（可执行程序）。
- 组织方式：顺序文件、索引文件、直接存取文件。
- 管理方式：文件系统通过目录结构和访问权限控制文件存储与访问 [[通用知识]]。

11. 文件目录的分类及查询方式

分类：

- 单级目录：所有文件集中管理，查询效率低。
- 二级目录：按用户划分目录，减少冲突。
- 树状目录：层次结构，支持多级嵌套（如 Linux 文件系统）[[通用知识]]。

查询方式：

- 线性搜索（单级目录）、哈希表加速（二级目录）、B+树索引（树状目录）[[通用知识]]。

12. 文件结构的三种主要组织方式及优劣对比

1. 顺序结构：
 - 优点：实现简单，适合连续读取。
 - 缺点：插入/删除效率低，易产生碎片。
2. 链接结构：
 - 优点：支持动态扩展，无碎片。
 - 缺点：随机访问慢，指针占用额外空间。
3. 索引结构：
 - 优点：支持快速随机访问，文件扩展灵活。
 - 缺点：索引表管理开销大，可能占用较多内存 [[通用知识]]。

13. 内存与外存的存储分配方式及异同

内存分配方式：动态分区（连续分配）、分页（非连续）。

外存分配方式：连续分配（如 FAT）、链接分配、索引分配（如 UNIX 文件系统）[[通用知识]]。

异同点：

- 相同点：均需解决空间分配和碎片问题。
- 不同点：
 - 内存分配以页或段为单位，需快速寻址；外存以块为单位，侧重减少磁盘 I/O。
 - 内存动态分区会产生外部碎片，而外存通过链表或索引消除碎片。
 - 内存分配依赖硬件（如 MMU），外存分配由文件系统软件管理 [[通用知识]]。

14. 提高磁盘 I/O 速度的途径

1. 缓冲技术：通过内存缓存减少实际磁盘访问次数。
2. 预读取：提前读取相邻数据块以利用局部性。
3. 优化调度算法：如使用 SCAN 算法减少寻道时间。
4. 磁盘高速缓存：利用磁盘内置缓存存储热点数据。
5. 文件布局优化：将频繁访问的数据连续存储。
6. RAID 技术：通过并行磁盘操作提升吞吐量 [[资料未直接提及，通用知识补充]]。